J. Symbolic Computation (1989) **8**, 553–580

Computing Primitive Elements of Extension Fields

KAZUHIRO YOKOYAMA

MASAYUKI NORO

TAKU TAKESHIMA

International Institute for Advanced Study of Social Information Science,

Fujitsu Limited,

140 Miyamoto, Numazu-shi, Shizuoka-ken, 410-03, Japan

(Received 10 December 1987)

Several mathematical results and new computational methods are presented for primitive elements and their minimal polynomials of algebraic extension fields. For a field $\mathbf{Q}(\alpha_1, ..., \alpha_t)$ obtained by adjoining algebraic numbers $\alpha_1, ..., \alpha_t$ to the rational number field $\mathbf{Q}$, it is shown that there exists at least one vector $\bar{s} = (s_1, ..., s_t)$ of integers in a specially selected set of $(t-1)N$ vectors such that $s_1\alpha_1 + s_2\alpha_2 + ... + s_t\alpha_t$ is a primitive element, where N is the degree of $\mathbf{Q}(\alpha_1, ..., \alpha_t)$ over $\mathbf{Q}$. Furthermore, a method is presented for directly calculating such a vector $\bar{s}$ that gives a primitive element. Finally, for a given polynomial f over $\mathbf{Q}$, a new method is presented for computing a primitive element of the splitting field of f and its minimal polynomial over $\mathbf{Q}$.

0747–7171/89/120553+28 $03.00/0

© 1989 Academic Press Limited

1. Introduction

For many applications in Computer Algebra, methods dealing with algebraic numbers are becoming more important. For example, the symbolic integration of a rational function requires operating in an extension field obtained as a subfield of the splitting field of its denominator. There are several approaches to describe extension fields in actual problems and applications. Sometimes algebraic numbers should be computed numerically or expressed in terms of radicals. But if we attempt to factor or integrate polynomials symbolically, we have to deal with extension fields in a precise, general and effective manner. One promising method is to describe extension fields of the rational field $\mathbf{Q}$ as polynomial factor rings. For an extension field K over $\mathbf{Q}$ generated by one algebraic number α, K is usually described as $\mathbf{Q}[x]/(f(x))$, where $f(x)$ is the minimal polynomial of α over $\mathbf{Q}$, i.e. a monic irreducible polynomial over $\mathbf{Q}$ which has α as a root. For more complicated extension fields generated by finitely many algebraic numbers, there have been few discussions about how to describe them.

In this paper such extension fields are considered. Since an extension field K generated by finitely many algebraic numbers has primitive elements α's, i.e. K is also a simple extension field $\mathbf{Q}(\alpha)$, it is natural to describe K as $\mathbf{Q}(\alpha)$. Here, we employ such description and discuss methods for finding primitive elements. Trager(1976) and Loos(1982b) discussed this approach and presented methods for finding primitive elements. Trager(1976) presented an algorithm which computes a primitive element and its minimal polynomial for extension fields generated by two elements. He also presented an algorithm for splitting fields. Loos(1982b) discussed the problem where an algebraic number is specified as a root of a square-free polynomial that lies in an isolating interval, and presented an algorithm for such a problem. Their methods for finding primitive elements and computing their minimal polynomials follow essentially the same idea. Their methods are based on the results of Kronecker and van der Waerden. (See van der Waerden(1931).) Resultants of polynomials are used for their actual computations of minimal polynomials. Furthermore, their methods are classified as *trial method* in this paper, because they choose a candidate and test whether it is really a primitive element or not. Their works provided a basic so-

lution to the posed problem. However, several questions are left. Our first question is on the bound of the number of trials. A bound for extension by adjoining two elements was given by Landau(1985). But the bound should be analyzed more precisely. Our second question is concerned with heuristic property of the algorithm, *i.e.*, *trial-and-test*. Is there any algorithm without *trial* ? The third is concerned with the specification or definition of extension fields. That is, how can we treat extension fields when they are specified by a set of algebraic equations? Of course, there are many other important questions and problems that will arise in many respective situations for dealing with extension fields. We discuss the first three questions in this paper. Following the contributions of Trager and Loos, we present new methods as well as mathematical basis for computing a primitive element of a field generated by two or more algebraic numbers, and for the splitting field of a polynomial over $\mathbf{Q}$.

The remainder of this paper is organized as follows. In Section 2 we discuss description of finite extension fields. In Section 3 we review presently known methods to compute minimal polynomials of algebraic integers over $\mathbf{Q}$. In Section 4 we discuss how to find primitive elements and present several new results. In Section 5 we discuss non-trial methods for splitting fields. Brief summary is given in the last section.

2. Description of finite extension fields

In dealing with algebraic numbers over $\mathbf{Q}$ on computers, it is important how to define them on computers. Some are defined in terms of radicals such as $1+\sqrt{2}$, some are defined as roots of polynomials, and others are defined as rational functions in algebraic numbers which are already defined. Anyway, for Computer Algebra, algebraic numbers are treated as symbols or variables with constraints. Thus a finite extension field is conveniently described as a polynomial ring over $\mathbf{Q}$ with generators as variables. To be exact, an extension field is described as a polynomial factor ring modulo an ideal associated with algebraic relations among algebraic numbers. Such a description will be adopted.

2.1. Extension generated by one element

An algebraic number α is defined by its minimal polynomial f_α. Then, $\mathbf{Q}(\alpha)$ is

identified with $\mathbf{Q}[x]/(f_\alpha(x))$ as fields, where $(f_\alpha(x))$ is the ideal generated by f_α. By this identification, the arithmetic, i.e. the operations of addition, subtraction, multiplication and division, can be done on computers as arithmetic in the polynomial ring $\mathbf{Q}[x]$ with reduction modulo $f_\alpha(x)$. See Loos(1982b) for details. We note that since $\mathbf{Q}[x]$ is a Euclidean ring, Euclidean reduction algorithm is applicable directly for computations of remainders by ideals.

2.2. Extension generated by two or more elements

Let $\alpha_1, \alpha_2, ..., \alpha_t$ be all distinct algebraic numbers over $\mathbf{Q}$ and let E be an algebraic extension field generated by $\alpha_1, \alpha_2, ..., \alpha_t$, i.e. $E = \mathbf{Q}(\alpha_1, \alpha_2, ..., \alpha_t)$. The definition of E depends on the definitions of $\alpha_1, ..., \alpha_t$. Generally, $(\alpha_1, ..., \alpha_t)$ is given as a zero of a system of t-variate algebraic equations. We provide the notion of *types* of definitions of extension fields.

Definition 2.1. The definition of E is called of *separable-type* if by changing the order of $\alpha_1, ..., \alpha_t$, each generator α_i is given as a root of a polynomial $g_i(x_i)$ over $\mathbf{Q}(\alpha_1, ..., \alpha_{i-1})$, i.e. $g_i(x_1, ..., x_i) = 0$ exists in the given system of algebraic equations, where $g_i(x_1, ..., x_i)$ is $g_i(x_i)$ with x_j's substituted for α_j's for $j < i$. Otherwise, the definition of E is called of *mixed-type*. Moreover, E is said to be *well-defined* if $E \cong \mathbf{Q}[x_1, ..., x_t]/I$, where I is the ideal generated by all polynomials appearing in the given system.

For the case where E is well-defined, E is determined uniquely up to an isomorphism. But for the case where E is not well-defined, it is possible that there are several fields, say candidates for E, which are not isomorphic to each other. Candidates are obtained as indecomposable submodules of $R = \mathbf{Q}[x_1, ..., x_t]/I$. Loos(1982b) eliminated this ambiguity by introducing the *isolating interval*.

The arithmetic on E can be performed as arithmetic on a multi-variate polynomial ring with a reduction modulo the ideal. But this arithmetic is rather complicated, since the Euclidean reduction algorithm cannot be applied directly for reduction modulo the ideal. So simple arithmetic is needed as in $\mathbf{Q}(\alpha)$. It is well-known that a finite extension field K of $\mathbf{Q}$ has its primitive element γ, i.e. $K = \mathbf{Q}(\gamma)$. Then if we have the following

two algorithms, namely one which computes a primitive element γ of E and its minimal polynomial over $\mathbf{Q}$, and one by which each α_i can be represented as polynomials in γ over $\mathbf{Q}$, then the arithmetic on E can be more efficiently performed on computers.

3. Minimal polynomials of algebraic numbers

In this section we review several presently known methods for computing minimal polynomials of algebraic numbers. These methods are utilized as parts of our new methods in Section 4 and Section 5. As stated in Section 2, the definition of an extension field is either of separable-type or of mixed-type, depending on which there is a method using the resultants of polynomials or a method using Gröbner bases.

3.1. Method using resultant

The method using resultant of polynomials is presented in Trager(1976) and Loos(1982b) for the case where the definition of K is of separable-type. In their method, resultant is used for computing *norm*.

Definition 3.1. Let F be a field of characteristic zero, and let E be a finite algebraic extension field of F. Moreover, let $\mathcal{E}$ be the set of all distinct embeddings of E into $\widetilde{F}$, where $\widetilde{F}$ is an algebraic closure of F. Then for an element $\alpha \in E$, the *norm* $\mathrm{N}_{E/F}(\alpha)$ of α from E to F is defined by

$$N_{E/F}(\alpha) = \prod_{g \in \mathcal{E}} \alpha^g.$$

The norm can be extended in a natural manner to polynomials with coefficients in E. For a polynomial h in $x, y, \ldots$ with coefficients in E, each embedding g of E into $\widetilde{F}$ can act on h by replacing coefficients of h by their conjugates by the action of g. By using this action, the *norm* $\mathrm{N}_{E/F}(h)$ is defined by

$$\mathrm{N}_{E/F}(h) = \prod_{g \in \mathcal{E}} h^g.$$

It is well-known that $\mathrm{N}_{E/F}(h)$ lies in $F[x, y, \ldots]$.

We note that if $F \subset E \subset D$ is a tower of finite extension fields, then $\mathrm{N}_{D/F} = \mathrm{N}_{E/F} \cdot \mathrm{N}_{D/E}$.

Now, the norm defined above can be used to compute minimal polynomials of elements in E. For an element β in E, let $f_0(x) = x - \beta$ and let $f(x) = N_{E/F}(f_0(x))$. Then f is

a monic polynomial over F and it has β as a root. Therefore, there exists the minimal polynomial f_β of β over F in the set of all irreducible factors of f over F. Moreover, we have the following well-known lemma. (See, e.g., Landau(1985).)

Lemma 3.1. (Well-known) *f is a power of the minimal polynomial f_β of β over F. If f is square-free, then $f = f_\beta$.*

By Lemma 3.1, the minimal polynomial f_α is obtained by computing the square-free part of f. Thus one way to compute minimal polynomials is to compute norms. From now on, we consider algebraic extension fields of $\mathbf{Q}$. So let F be either $\mathbf{Q}$ or an algebraic extension field of $\mathbf{Q}$. For actual computation of norm, resultant is used as follows.

Case 1. $E = F(\alpha)$

Let f_α be the minimal polynomial of α over F. For any element β in E, β can be expressed as a polynomial in α, say $\beta(\alpha)$. The norm $\mathrm{N}_{E/F}(x-\beta)$ can be computed as follows:

$$\mathrm{N}_{E/F}(x-\beta(\alpha)) = \mathrm{Res}_y(f_\alpha(y), x-\beta(y)).$$

The correctness of the above equation follows from Definition 2.1 and properties of resultant. Especially, if β is expressed as $s\alpha + t$, where $s, t \in F$, the norm $\mathrm{N}_{E/F}(x-\beta)$ is obtained by the following equation.

$$\mathrm{N}_{E/F}(x-\beta) = f_\alpha((x-t)/s)/k,$$

where k is the leading coefficient of $f_\alpha((x-t)/s)$.

Case 2. $E = F(\alpha_1, ..., \alpha_t)$

We consider the case where E is well-defined. By the definition of E, we can assume that the minimal polynomial g_{i+1} of α_{i+1} over $E_i = F(\alpha_1, ..., \alpha_i)$ is given for $0 \le i \le t-1$. Then by applying the method presented in Case 1 to extensions E_{i+1}/E_i repeatedly, we obtain the norm of β from E to F for $\beta \in E$ as follows:

Let $h_t = x - \beta(\alpha_1, ..., \alpha_t)$. For $i < t$, h_i is defined as $\mathrm{N}_{E_{i+1}/E_i}(h_{i+1})$, i.e.

$$h_i = \mathrm{Res}_{y_{i+1}}(g_{i+1}(y_{i+1}), h_{i+1}(x; \alpha_1, ..., \alpha_i, y_{i+1})).$$

Then h_0 is the norm $\mathrm{N}_{E/F}(x-\beta)$.

Remark. As for the case where E is not well-defined, if each α_i is defined by its minimal

polynomial f_i over F, then the similar method is applicable. We also define polynomials h_i^* as follows:

Let $h_t^* = x - \beta(\alpha_1, ..., \alpha_t)$. For $i < t$, $h_i^* = \mathrm{Res}_{y_{i+1}}(f_{i+1}(y_{i+1}), h_{i+1}^*(x; \alpha_1, ..., \alpha_i, y_{i+1}))$. Then h_0^* is a polynomial over F, and has the following property.

Lemma 3.2. *For each irreducible factor h of h_0^*, there are conjugates $\alpha_1', ..., \alpha_t'$ such that h is the minimal polynomials of $\beta(\alpha_1', ..., \alpha_t')$ over F, where α_i' is a conjugate of α_i over F. Conversely, for any conjugates $\alpha_1', ..., \alpha_t'$, h_0^* has the minimal polynomial of $\beta(\alpha_1', ..., \alpha_t')$ over F as an irreducible factor.*

Lemma 3.2 follows from the fact that h_0^* is expressed as $h_0^*(x) = \prod(x - \beta(\alpha_1', ..., \alpha_t'))$, where α_i' ranges over all conjugates of α_i by the definition of resultant. Hence, from Lemma 3.2, we can compute the minimal polynomial of $\beta(\alpha_1', ..., \alpha_t')$ over F by factoring h_0^*.

As for efficient algorithms for resultant, see Collins(1971) and Loos(1982a).

3.2. Method using Gröbner basis

For the case where the definition of K is of mixed-type, a method using Gröbner basis is applicable. By the studies on algebraic relations of polynomials and systems of algebraic equations, several methods for computing minimal polynomials are known. (See Arnon & Sederberg(1984), Gianni & Trager & Zacharias(1987) and Kobayashi(1987).) Here we review two methods for the well-defined case.

Let $R = \mathbf{Q}[x_1, ..., x_t]$, and let I be the ideal generated by $p_1, ..., p_s$, where p_i is a polynomial in $x_1, ..., x_t$ over $\mathbf{Q}$. Since $K = R/I$ is an algebraic extension field of $\mathbf{Q}$, there are algebraic integers α_i's associated with x_i's such that $K = \mathbf{Q}(\alpha_1, ..., \alpha_t)$. Consider the following polynomial ring R' and its ideal I'.

$R' = \mathbf{Q}[y, x_1, ..., x_t]$ and $I' = (p_1, ..., p_s, \beta - y)$, the ideal generated by $p_1, ..., p_s$ and $\beta - y$.

Let Γ be the reduced Gröbner basis with respect to the lexicographic order $x_t \triangleright \cdots \triangleright x_1 \triangleright y$ and with head coefficients 1. Then in Γ there exists exactly one polynomial, say $G(y)$, which does not contain variables $x_1, ..., x_t$. Then $G(y)$ is the minimal polynomial of β.

As a special case, we consider the case where an algebraic number β is expressed as $\sum_{i=1}^{t} a_i\alpha_i$, $a_i \in \mathbf{Q}$, and β is known to be a primitive element of K. In this case, the minimal polynomial h of β can be computed in the following manner.

Transform the basis $\{x_1, x_2, ..., x_t\}$ of the ring $\mathbf{Q}[x_1, x_2, ..., x_t]$ to $\{y_1, y_2, ..., y_t\}$, where $y_2 = x_2, ..., y_t = x_t$ and $y_1 = \sum_{i=1}^{t} a_i x_i$. Let $R' = \mathbf{Q}[y_1, y_2, ..., y_t]$ and $I' = (p'_1, ..., p'_s)$, where each p'_i is obtained from p_i as a polynomial in $y_1, y_2, ..., y_t$. Then in the reduced Gröbner basis Γ of I' with respect to the lexicographical order $y_1 \lhd \cdots \lhd y_t$, there exists a polynomial, say $G_1(y_1)$, which does not contain variables $y_2, ..., y_t$. Then $G_1(y_1)$ is the minimal polynomial of β. Moreover, for each $i \neq t$, there exists $y_i - G_i(y_t)$ in Γ for some polynomial h_i in y_t, and $\Gamma = \{G_1(y_1), y_2 - G_2(y_1), ..., y_t - G_t(y_1)\}$.

If $\beta = \sum_{i=1}^{t} a_i\alpha_i$, $a_i \in \mathbf{Q}$, is not a primitive element, the reduced Gröbner basis Γ does not have such a form. Therefore we can apply this method for testing primitiveness of elements. (See Algorithm 4.3.)

As for the case where K is not well-defined, we have to decompose the polynomial factor ring R/I, i.e., we have to decompose the ideal I. In this case, similar methods as above exist. (See Gianni & Trager & Zacharias(1987) and also Kobayashi & Moritsugu & Hogan(1988).)

Remark.

Other methods, which are based on linear equations over $\mathbf{Q}$-vector space R, are used in Yokoyama & Noro & akeshima(1987) for the separable-type case, and in Kobayashi & Moritsugu & Hogan(1988) for the mixed-type case.

4. Finding primitive elements.

In this section we consider algebraic extension fields over $\mathbf{Q}$ described in Section 2, and discuss several methods to find primitive elements of these fields.

Let $\alpha_1, ..., \alpha_t$ be all distinct algebraic numbers over $\mathbf{Q}$, and let K be an algebraic extension field generated by these numbers. Let f_i be the minimal polynomial of α_i over $\mathbf{Q}$, and n_i =deg f_i for $1 \leq i \leq t$. Moreover, let $\mathcal{E}$ be the set of all distinct embeddings of K into $\widetilde{\mathbf{Q}}$, where $\widetilde{\mathbf{Q}}$ is an algebraic closure of $\mathbf{Q}$. Then $|\mathcal{E}| = |K{:}\mathbf{Q}|$. So let $N = |K{:}\mathbf{Q}|$. We define the following.

Let $\mathbf{Z}^t$ be the direct product of $\mathbf{Z}$ with itself t times. For an element $\bar{s} = (s_1, ..., s_t)$ in $\mathbf{Z}^t$, an algebraic number $\alpha(\bar{s})$ is defined as

$$\alpha(\bar{s}) = s_1\alpha_1 + s_2\alpha_2 + ... + s_t\alpha_t.$$

Moreover, for simplicity, we write $\alpha(\bar{s})^g$ as the image of $\alpha(\bar{s})$ by the action of $g \in \mathcal{E}$. Then $\alpha(\bar{s})^g$ is expressed as

$$s_1\alpha_1^g + s_2\alpha_2^g + ... + s_t\alpha_t^g.$$

It is known that there exist primitive elements of K in the set $\{\alpha(\bar{s})|\bar{s} \in \mathbf{Z}^t\}$. (See Lang(1976).) We call a vector $\bar{s}$ which gives a primitive element a *generating vector* for K. Our problem is to find a generating vector. To obtain such a vector, we have two kinds of methods, *trial method* and *non-trial method.* In the trial method, we repeatedly take a vector and obtain a candidate for a primitive element and then test whether it is really a primitive element or not, until eventually we get a primitive element. In the non-trial method, which we propose in this paper, we compute a vector which is known to be a generating vector, and then make up a primitive element.

Trial methods have been presented by Trager(1976) and Loos(1982b). Their methods are applicable to any algebraic extension field whose definition is of separable-type, but inapplicable to one whose definition is of mixed-type. The reason is that resultant cannot be used to obtain norms for the mixed-type case. For the mixed-type case, however, there exists another trial method, in which we employ Gröbner basis for testing primitiveness and computing the minimal polynomials. (See Section 3.2.)

In the following, we first consider trial methods, and discuss how many trials are

needed in each computation. We next present a new non-trial method. Finally we give the comparison of efficiency among methods described here.

4.1. Finding primitive elements by trial methods

We will show that the number of trials in trial methods is bounded by $(t-1)N$. We begin with Lemma 4.1 and Theorem 4.2.

Lemma 4.1. (Well-known) *$\alpha(\bar{s})$ is a primitive element if and only if $\alpha(\bar{s})^g \neq \alpha(\bar{s})^h$ for any pair of distinct embeddings g and h.*

Let V be the set of all vectors $\bar{s}$ in $\mathbf{Z}^t$ which are not generating vectors, i.e. $\alpha(\bar{s})$'s are not primitive elements of K. Then by the definition of $\alpha(\bar{s})$ and Lemma 4.1, V is expressed as follows.

$V = \{\bar{s} \in \mathbf{Z}^t \mid \alpha(\bar{s})^g = \alpha(\bar{s})^h$ for some pair of distinct embeddings g and h $\}$.

Let $V(g,h) = \{\bar{s} \in \mathbf{Z}^t \mid \alpha(\bar{s})^g = \alpha(\bar{s})^h \}$. Then $V = \cup_{g \neq h \in \mathcal{E}} V(g,h)$.

Theorem 4.2. $V = \cup_{g \in \mathcal{E}-\{1\}} V(1,g)$.

Proof. Consider the Galois closure $\tilde{K}$ of K in $\tilde{\mathbf{Q}}$ and its Galois group $\mathcal{G}$. Then $\mathcal{E}$ can be considered as the set $\mathcal{G}/\mathcal{G}_K$ of all $\mathcal{G}_K$-cosets, where $\mathcal{G}_K$ is the subgroup of $\mathcal{G}$ consisting of all elements each of which fixes all numbers in K. Therefore for each $g \in \mathcal{E}$, there is some $g' \in \mathcal{G}$ such that $\alpha_i^g = \alpha_i^{g'}$. And for such g', it follows that $\alpha_i^g = \alpha_i^h$ for $h \in \mathcal{G}_K g'$. From this, we consider each element of $\mathcal{E}$ as an element of $\mathcal{G}$. Then by using properties of group, we have the following.

$\alpha(\bar{s})^g = \alpha(\bar{s})^h$ if and only if $\alpha(\bar{s}) = \alpha(\bar{s})^{hg^{-1}}$.

Hence we have $V = \cup_{g \in \mathcal{G}-\{1\}} V(1,g)$. Since there is an element in $\mathcal{E}$ corresponding with each $\mathcal{G}_K$-coset, we obtain the following.

$V = \cup_{g \in \mathcal{G}-\{1\}} V(1,g) = \cup_{\mathcal{G}_K g \in \mathcal{G}/\mathcal{G}_K - \{\mathcal{G}_K\}} V(1, \mathcal{G}_K g) = \cup_{g \in \mathcal{E}-\{1\}} V(1,g)$. Q.E.D.

From now on, we use $V(g)$ instead of $V(1,g)$ for $g \in \mathcal{E} - \{1\}$. Consider an element $\bar{s}$ in $V(g)$. Then by definition, $\bar{s} = (s_1, ..., s_t)$ satisfies the following equation.

$s_1\alpha_1 + s_2\alpha_2 + ... + s_t\alpha_t = s_1\alpha_1^g + s_2\alpha_2^g + ... + s_t\alpha_t^g$.

By considering the above equation as a linear equation over $\mathbf{Q}$, $V(g)$ is a $\mathbf{Z}$-submodule of the solution of this linear equation. Since g is not the identity embedding, the rank of

$V(g)$ as a $\mathbf{Z}$-module is at most $t-1$. Note that $|V(g)|$ is infinite, when the rank is not equal to 0. Consider the case $t=2$, then the rank of $V(g)$ is at most 1. Therefore there is at most one integer s_g such that $(1, s_g)$ belongs to $V(g)$. As $|\mathcal{E}-\{1\}| = N-1$, there are at most $N-1$ elements $\bar{s}=(1,s)$ in V. Hence we have the following directly from this argument.

Theorem 4.3. *There is at least one integer s among N distinct integers such that $\alpha_1+s\alpha_2$ is a primitive element of $K=\mathbf{Q}(\alpha_1,\alpha_2)$, where N is the degree of K over $\mathbf{Q}$.*

Remark. For the problem of factoring polynomials over algebraic extension fields, Landau(1985) presented a bound for the number of necessary trials to obtain irreducible factors. When we apply her results (Lemma 1.6 in Landau(1985)), the number of trials for finding a primitive element of $\mathbf{Q}(\alpha_1,\alpha_2)$ is bounded by $N(N-1)/2$. Hence Theorem 4.3 gives an improved bound over Landau's result. In Section 4.3, we give further discussion.

Next, we discuss the case for arbitrary t. Let $K_0=\mathbf{Q}$, $K_i=\mathbf{Q}(\alpha_1,...,\alpha_i)$ and $N_i = |K_i : \mathbf{Q}|$, for $1 \le i \le t$. Moreover, let $\bar{s}(i)=(s_1,...,s_i,0,...,0)$ for $\bar{s}=(s_1,...,s_t) \in \mathbf{Z}^t$ and $1 \le i \le t$. For a vector $\bar{u}=(u_1,...,u_k)$ in $\mathbf{Z}^k$, $k \le t$, u can be considered as a vector $(u_1,...,u_k,0,...,0)$ in $\mathbf{Z}^t$, and so $\alpha(\bar{u})$ can be defined. Similarly as V, V_i can be defined for K_i and $\bar{s}(i)$. Since the set of all distinct embeddings $\mathcal{E}_i$ of K_i into $\widetilde{\mathbf{Q}}$ is the set of all distinct restriction of elements of $\mathcal{E}$ on K_i, for an element $\bar{s}(i)$ in V_i there is an embedding g in $\mathcal{E}$ such that $\mathcal{E}$ is not identical on K_i and $\alpha(\bar{s}(i)) = \alpha(\bar{s}(i))^g$. From this, it follows that if $\bar{s}(i)$ does not belong to V_i for any i, then $\bar{s}$ does not belong to V. From this observation, we can always find each generating vector for K_i successively in such a way that we determine $u \in \mathbf{Z}$ by trial-and-test such that $(s_1, s_2, ..., s_{i-1}, u, ..., 0)$ is a generating vector for K_i, after getting $\bar{s}(i-1)=(s_1,...,s_{i-1},0,...,0)$ as a generating vector for K_{i-1}. We call this procedure *successive-trial.* Then we have the following by Theorem 4.3.

Theorem 4.4. *If we employ successive-trial, the number T of trials is bounded as follows:*

$$T \le N_2 + N_3 + ... + N_t < (t-1)N.$$

For the case where the definition of K is of separable-type, we can employ successive-trial. But for the case where the definition of K is of mixed-type, we cannot employ

successive-trial directly. But by applying methods discussed in Section 3.2, we can obtain the minimal polynomials of α_i's. From this, we can transform the mixed-type case into the separable-type case. However, there exists another trial method (Algorithm 4.3) applicable directly to the mixed-type case, in which the number of trials is bounded by the same order $(t-1)N$ as in successive-trial. Before we show a new trial method, we provide necessary notions and theorems.

Definition 4.1. Let S be a subset of $\mathbf{Z}^t$. S is called a *t-base set*, if any distinct t-elements of S are linearly independent over $\mathbf{Q}$.

By well-known results on linear algebra and by the fact that $\mathbf{Q}$ is an infinite field, for any positive integer $m \geq t$ there are infinitely many t-base sets consisting of m elements. Moreover, a t-base set can be obtained whose elements are expressed in terms of integer parameters. We will show later several t-base sets obtained by one parameter.

Let S be a t-base set consisting of $(t-1)\times(N-1)+1$ elements. Then the following theorem holds.

Theorem 4.5. *In S there is at least one element $\bar{s}$ which is a generating vector, i.e. $\alpha(\bar{s})$ is a primitive element of K.*

Proof. By Theorem 4.2, $V = \cup_{g\in\mathcal{E}-\{1\}} V(g)$ and rank $V(g) \leq t-1$ for $g \in \mathcal{E}-\{1\}$. Let $V^*(g)$ be the vector space over $\mathbf{Q}$ spanned by $V(g)$. We show that S is not contained in V. Assume the contrary. Then in a set of at most $(t-1)\times(N-1)$ elements of S there is a basis of $V^*(g)$ for $g \in \mathcal{E}-\{1\}$. Since S has $(t-1)\times(N-1)+1$ elements, there is at least one element $\bar{s}$ which does not belong to any $V^*(g)$ by the definition of t-base set. This contradicts the assumption. Hence we conclude that S is not contained in V. Q.E.D.

By Theorem 4.5, if we seek a generating vector $\bar{s}$ from a t-base set S of $\mathbf{Z}^t$, the number of trials T is bounded as follows;

$$T \leq (t-1)\times(N-1)+1 < (t-1)N.$$

This bound has the same order as the bound for the successive-trial method. We call trials from a t-base set *t-base-trial.*

Here we remark the size of search space of generating vectors. By using similar arguments as in the proof of Theorem 4.3, we also have the following.

Theorem 4.6. *Let S_i be a set consisting of N_i distinct integers. Then there is at least one element in $\{1\} \times S_2 \times \cdots \times S_t$ which is a generating vector for K.*

From Theorem 4.6, the search space is included in $\{1\} \times S_2 \times \cdots \times S_t$, and the size is bounded by $N_2 \times \cdots N_t$. Moreover, it might happen that for a badly chosen integer u, no vector $\bar{s}$ with $s_2 = u$ can be a generating vector. Therefore, if we do not employ t-base-trial, we can only assure that the number of trials T is bounded as follows:

$T < N_2 \times \cdots \times N_t < N^{t-1}$.

In the following, we present examples of t-base sets, which are expressed in terms of one parameter. Consider a vector $\bar{s}$ obtained by using a parameter u. Then each component s_i is a function of u. Here we consider the case where s_i is a rational function of u. So we write $s_i(u)$ and $\bar{s}(u)$ instead of s_i and $\bar{s}$ respectively. Let $\mathrm{M}(u_1, ..., u_t)$ be a matrix whose $j-$th row is a vector $\bar{s}(u_j)$, and let $D(u_1, ..., u_t)$ be the determinant of $\mathrm{M}(u_1, ..., u_t)$. Then we have the following lemma which can be shown easily.

Lemma 4.7. *If $D(u_1, ..., u_t) \neq 0$ for any distinct t integers $u_1, ..., u_t$, then $\{\bar{s}(u) | u \in \mathbf{Z}\}$ is a t-base set.*

Using Lemma 4.7, we have the following examples for t-base sets.

Examples.

1. Type 1.

Define a vector $\bar{s}(u)$ for an integer u as follows:

$\bar{s}(u) = (1, u, u^2, ..., u^{t-1})$.

Then $\{\bar{s}(u) | u \in \mathbf{Z}\}$ is a t-base set, since $\mathrm{M}(u_1, ..., u_t)$ is a Vandermonde matrix.

2. Type 2.

Fix distinct t integers $v_1, ..., v_t$. Define a vector $\bar{s}(u)$ for an integer u as follows:

$\bar{s}(u) = k \times (1/(u - v_1), 1/(u - v_2), ..., 1/(u - v_t))$,

where k is an integer such that $\bar{s}(u)$ is a vector over $\mathbf{Z}$. Then $\{\bar{s}(u) | u \in \mathbf{Z} - \{v_1, ..., v_t\}\}$ is a t-base set. Because, for any distinct t integers $u_1, ..., u_t$ in $\mathbf{Z} - \{v_1, ..., v_t\}$, it follows that

$D(u_1, ..., u_t) = (-1)^{t(t-1)/2} P(v_1, ..., v_t) P(u_1, ..., u_t) / \prod_{i,j}(v_i - u_j)$,

where P is the determinant of a Vandermonde matrix, and hence $D(u_1, ..., u_t) \neq 0$ for any distinct t integers $u_1, ..., u_t$ in $\mathbf{Z} - \{v_1, ..., v_t\}$.

4.2. Constructing primitive elements by a non-trial method

We will show that we can obtain deterministically a generating vector $\bar{s}$ for K. We define the bounds of algebraic numbers as roots of polynomials over $\mathbf{Q}$. The algebraic closure $\widetilde{\mathbf{Q}}$ can be embedded in the complex number field $\mathbf{C}$. Therefore absolute values are defined for elements of $\widetilde{\mathbf{Q}}$. By using absolute values, the following bounds are defined for algebraic numbers $\alpha_1, ..., \alpha_t$.

Definition 4.2. Let b_i be a rational number which is greater than the absolute values of any conjugates of α_i over $\mathbf{Q}$, and let c_i be a positive rational number whose reciprocal is not greater than the absolute values of the differences of any two distinct conjugates of α_i over $\mathbf{Q}$.

The above bounds b_i and c_i can be computed by using the minimal polynomial of α_i over $\mathbf{Q}$. We discuss this later. (See Remark.)

Consider a vector $\bar{s} = (s_1, ..., s_t) \in \mathbf{Z}^t$ which is defined as $s_1 = 1$ and $s_i \geq 2(s_1 b_1 + s_2 b_2 + ... + s_{i-1} b_{i-1}) c_i$ for $i \geq 2$.

Then we have the following Theorem.

Theorem 4.8. *$\bar{s}$ is a generating vector for K, i.e. $\alpha(\bar{s})$ is a primitive element of K.*

Proof. By Theorem 4.2, we have only to show that $\alpha(\bar{s}) \neq \alpha(\bar{s})^g$ for $g \in \mathcal{E} - \{1\}$. Assume, to the contrary, that there is an embedding g in $\mathcal{E} - \{1\}$ such that $\alpha(\bar{s}) = \alpha(\bar{s})^g$. Let j be the largest integer such that $\alpha_j \neq \alpha_j^g$. Then it follows that

$s_1(\alpha_1 - \alpha_1^g) + ... + s_{j-1}(\alpha_{j-1} - \alpha_{j-1}^g) = s_j(\alpha_j^g - \alpha_j)$.

By using bounds, the left-hand side is bounded as follows;

$|s_1(\alpha_1 - \alpha_1^g) + ... + s_{j-1}(\alpha_{j-1} - \alpha_{j-1}^g)| \leq s_1|\alpha_1 - \alpha_1^g| + ... + s_{j-1}|\alpha_{j-1} - \alpha_{j-1}^g| < 2s_1 b_1 + ... + 2s_{j-1} b_{j-1}$.

The right-hand side is also bounded as follows;

$|s_j(\alpha_j^g - \alpha_j)| \geq s_j / c_j$.

Hence $2s_1b_1 + ... + 2s_{j-1}b_{j-1} > s_j/c_j$, and this contradicts the definition of s_j. Q.E.D.

Let b be the largest number among b_i's, and let c be the largest number among c_i's. Moreover, let u be the smallest integer such that $u \geq 4bc$ and $u \geq 2$. We also define a vector $\bar{u} = (1, u, u^2, ..., u^{t-1})$. Then we have the following Theorem by the similar proof as in Theorem 4.8.

Theorem 4.9. *$\bar{s} = (1, u, u^2, ..., u^{t-1})$ is a generating vector for K, i.e. $\alpha(\bar{u})$ is a primitive element of K.*

Remark. For the case where minimal polynomials f_i of α_i over $\mathbf{Q}$ are known, the bounds b_i and c_i can be computed by the following way by using Mignotte(1982)'s result.

For a polynomial $h(x) = \sum_{i=0}^{d} h_i x^i$, let $\| h \| = (\sum_{i=0}^{d} h_i^2)^{1/2}$ and let $|h| = max\{|h_0|, ..., |h_d|\}$. Then the following holds.

$b_i \geq min\{|f_i| + 1, \| f_i \|\}$,

$c_i \geq min\{(2b_i)^{n_i(n_i-1)/2-1}/|D_i|^{1/2}, n_i^{(n_i+2)/2} \| f_i \|^{n_i-1} / \sqrt{3}|D_i|^{1/2}\}$,

where D_i is the discriminant of f_i and n_i=deg f_i.

We note that there is a way to determine b_i and c_i numerically.

By Theorem 4.8, Theorem 4.9 and above Remark, we have a non-trial method for the case where the definition of k is of separable-type. Thus we can find the primitive elements in a deterministic manner. Further study is needed for the case when the above bounds, and hence coefficients of the resulting minimal polynomial of the primitive element, become very large.

4.3. Algorithms and remarks

Here we present algorithms for finding primitive elements and discuss their efficiency. As a typical case, we consider the case $t = 2$. As mentioned before, Trager(1976) and Loos(1982b) presented algorithms using norms for testing candidates. Algorithm 4.1 is essentially the same as their algorithms.

Output: a primitive element γ of $\mathbf{Q}(\alpha, \beta)$ and its minimal polynomial g over $\mathbf{Q}$.

$s := 1$;

$h(x) :=\mathrm{Res}_y(f(y), g(x - sy))$;

While $\mathrm{GCD}(h(x), (dh/dx)(x)) \neq 1$ do

$\{s := s + 1$;

$h(x) :=\mathrm{Res}_y(f_1(y), g_2(x - sy)\}$;

Return $\gamma = s\alpha + \beta$ and $h(x)$.

Algorithm 4.2. (non-trial method for well-defined case)

Input: same as in Algorithm 4.1.

Output: same as in Algorithm 4.1.

compute b_β;

compute c_α;

$s := \lceil 2b_\beta c_\alpha \rceil$;

$h(x) :=\mathrm{Res}_y(f(y), g(x - sy))$;

Return $\gamma = s\alpha + \beta$ and $h(x)$.

In the above, $\lceil a \rceil$ denotes the smallest integer greater than or equal to the rational a.

The complexity of the above algorithms is evaluated as follows. For the sake of simplicity, assume that degrees of polynomials f and g are both n, and that $f(x)$ and $g(x, y)$ are polynomials over $\mathbf{Z}$, where $g(x, y)$ is $g(x)$ with y substituted for α_1. Let M be the smallest integer which exceeds all the coefficients of $f(x)$ and $g(x, y)$. Then the integer coefficients in $g'(x, y) = g(x - sy, y)$ are bounded by $n^2 s^n 2^n M$. Therefore semi-norms $\|f\|$ and $\|g'\|$ are bounded by $n^3 s^n 2^n M$, where the semi-norm of a polynomial is the square root of the sum of the square of all the coefficients. In the modular algorithm for resultant (see Loos(1982a)), the time required to compute resultant for two 2-variate polynomials with degrees n is $O(n^5 L(d) + n^4 L(d)^2)$, where $L(d)$ denotes the maximal length of the semi-norms. Thus, the computation of resultant takes no more than $O(n^5 L(max\{\|f\|, \|g'\|\}) + n^4 L(max\{\|f\|, \|g'\|\}))$-steps. And the length of the semi-norm $\|h\|$ of resulting polynomial h is bounded by $O(nL(max\{\|f\|, \|g'\|\})$. In Algorithm 4.1, we have also to compute $\mathrm{GCD}(h, dh/dx)$. Since GCD of polynomials with degrees m and semi-norms d can

be computed in $O(m^3L(d)^2)$ by the modular P.R.S. algorithm (also see Loos(1982a)), GCD$(h, dh/dx)$ can be computed in $O(n^8L(max\{\|f\|, \|g'\|\})$. Since the number of trials is bounded by n^2 by Theorem 4.3, Algorithm 4.1 computes a primitive element and its minimal polynomial in $O(n^{10}L(max\{\|f\|, \|g'\|\})^2)$-steps and so in $O(n^{10}L(n^3s^n2^nM)^2)$-steps. On the other hand, in Algorithm 4.2, if we choose $|g|+1$ as b_β, b_β is bounded by nM^n. Moreover, c_α is bounded by $n^{(n+3)/2}M$, if we choose $n^{(n+2)/2}\|f\|/(3D)^{1/2}$ as c_α. Then s is bounded by $2n^{2n}M^2$ (an over-estimated value), and it is computed in $O(n^3L(n^{1/2}M)+n^2L(n^{1/2}M)^2)$-steps. Thus, Algorithm 4.2 computes a primitive element and its minimal polynomial in $O(n^5L(n^{2n^2+2}M^{2n+1}2^{2n})+n^4L(n^{2n^2+2}M^{2n+1}2^{2n}))^2)$-steps. Hence the cost of Algorithm 4.2 is much less than the cost to compute GCD$(h, dh/dx)$ in Algorithm 4.1. So even if the first trial succeeds in Algorithm 4.1, Algorithm 4.2 is faster than Algorithm 4.1.

The above algorithms do not compute the representations of α and β by the obtained primitive element γ. From his experience, Loos(1982b) pointed out that 80 % of the whole computation time is consumed in such representation stage of his algorithm. In his algorithm, the representations are obtained in a usual manner, *i.e.*, by computing GCD$(g''(x), f(x))$ over the extension field $\mathbf{Q}(\gamma)$, where $g''(x) = g(\gamma - sx)$. According to Landau(1985), the cost to compute such GCD is bounded by $O(n^{11}log^2(\|h\|)log^2(max\{\|f\|, \|g''\|\}))$-steps. This overwhelms even the bound for Algorithm 4.1. Moreover, the bound contains s in $\|h\|$, and we cannot present better bound for s presented above in our algorithm. This will affect the bound for representation stage. Although the bounds for s are not tight, it is somewhat discouraging. The presented algorithm, however, will serve us if any faster GCD-algorithm over algebraic extension fields is invented, or if we can get the representations by any other method faster than usual GCD method.

Remark. When β is given by the minimal polynomial $\tilde{g}$ over $\mathbf{Q}$ instead of g over $\mathbf{Q}(\alpha)$, we have to factorize $h(x) = \mathrm{Norm}_{K/\mathbf{Q}}(\tilde{g}(x-s\alpha))$ to eliminate ambiguity of extension fields. In this case, if $h(x)$ is square-free, each irreducible factor gives the minimal polynomial of a primitive element of each field generated by conjugates of α and β. If a trial method is employed, the number of trials is bounded as follows by the similar arguments as in

Theorem 4.3.

Theorem 4.10. *There is at least one integer s among $n_1 n_2(n_2-1)$ distinct integers such that $h(x)$ is square-free for such s, i.e. each irreducible factor of $h(x)$ gives the minimal polynomial of a primitive element.*

Theorem 4.10 is an improvement of Lemma 1.6 in Landau(1985). As for a non-trial method, we have the following by the same argument as in the proof of Theorem 4.8.

Theorem 4.11. *Let $s = 2b_\beta c_\alpha$, where b_α and c_β are defined in Definition 4.2. Then $h(x)$ is square-free.*

We note that these result can be applied directly to factorization of polynomials over algebraic extension field.

Next we consider the general case $t \geq 3$. When the definition of K is of separable-type, we can apply algorithms for $t = 2$ successively. At each step, we can employ either a trial method or a non-trial method. Thus we consider the other case where the definition of K is of mixed-type and well-defined. We show a new trial method using t-base trial. In this method, affine transformation of the basis described in Section 3.2 is employed to test whether a candidate is a primitive element or not. This method includes the representations of each α_i in terms of the primitive element.

Algorithm 4.3. (t-base-trial method for mixed-type case)

Input: the polynomials $h_1,....,h_m$ which generate a maximal ideal I in $\mathbf{Q}[x_1,...,x_n]$, i.e. $K \cong \mathbf{Q}[x_1,...,x_n]/I$, and a t-base set which is expressed in term of a parameter u.

Output: a generating vector $\bar{s}$, the minimal polynomial of $\alpha(\bar{s})$ over $\mathbf{Q}$ and the representations of α_i's.

$u := 1$;

compute a vector $\bar{s}(u)$;

$y := s_1(u)x_1 + \cdots s_n(u)x_n$;

transform the basis $\{x_1, ..., x_n\}$ to $\{y, x_2, ..., x_n\}$,

i.e. compute $h_i'(y, x_2, ..., x_n)$ for each h_i;

let $I' = (h_1', ..., h_m')$;

compute Gröbner basis Γ of I' w. r. t. the lexicographical order $y \lhd x_2 \lhd \cdots \lhd x_n$;

While (Γ is not good*1)) do

$\{u := u + 1$;

compute a vector $\bar{s}(u)$;

$y := s_1(u)x_1 + \cdots s_n(u)x_n$;

transform the basis $\{x_1, ..., x_n\}$ to $\{y, x_2, ..., x_n\}$;

let $I' = (h_1', ..., h_m')$;

compute Gröbner basis Γ of I w. r. t. the lexicographical order $y \lhd x_2 \lhd \cdots \lhd x_n\}$;

Return $\bar{s}(u)$ and Γ.*2)

*1) Γ is good if it has the form $\{G_1(y), x_2 - G_2(y), ..., x_n - G_n(y)\}$.

*2) $G_1(y)$ gives the minimal polynomial of $\alpha(\bar{s}(u))$, α_i is represented by $G_i(y)$ for $2 \leq i \leq n$ and α_1 is represented by $(y - s_2(u)G_2(y) - ... - s_n(u)G_n(y))/s_1(u)$.

We do not discuss the complexity of Algorithm 4.3, since exact complexity of the computation of Gröbner basis is not known. We can assert only that the number of trials is bounded by $(t-1)d_1 \cdots d_n$, where d_1,...,d_n are the largest n total degrees of h_1,...,h_s. This bound follows from Theorem 4.5 and Bézout's theorem, that is, $N \leq d_1 \cdots d_n$.

5. Splitting fields of polynomials

For a polynomial $f(x)$ over $\mathbf{Q}$, the splitting field K is obtained by adjoining all roots of f to $\mathbf{Q}$. We describe a non-trial algorithm by using *generating vectors*. For simplicity, let $f(x)$ be a monic irreducible polynomial over $\mathbf{Q}$. The irreducibility can be assumed, since the splitting field of a polynomial can be obtained from all the splitting fields of its factors.

Let $\alpha_1, ..., \alpha_n$ be all the roots of f, where $n = \deg_x f(x)$, and let K be the splitting field of f, i.e. $K = \mathbf{Q}(\alpha_1, .., \alpha_n)$, and let $S = \{\alpha_1, ..., \alpha_n\}$. We note that since $\alpha_1 + ... + \alpha_n \in \mathbf{Q}$, K is generated by $n-1$ distinct roots of f. Similarly as in Section 4, we define the following. For a vector $\bar{s} = (s_1, ..., s_n)$ in $\mathbf{Q}^n$, we define $\alpha(\bar{s})$ as $\alpha(\bar{s}) = s_1\alpha_1 + s_2\alpha_2 + ... + s_n\alpha_n$. Moreover, let $\bar{s}(i)$ denote the vector $(s_1, ..., s_i, 0, ..., 0)$.

We define *generating vectors* for splitting fields, which are slightly extended from those for arbitrary algebraic extension fields in Section 4.

Definition 5.1. A vector $\bar{s} = (s_1, ..., s_n)$ is called a *generating vector* for a primitive element if for any ordering of roots of f, $\alpha(\bar{s}(i)) = \alpha((s_1, s_2, ..., s_i, 0, ..., 0))$ is a primitive element of $\mathbf{Q}(\alpha_1, \alpha_2, ..., \alpha_i)$ for $1 \leq i \leq n$.

In the following we show three examples of generating vectors, two of which are applicable to special cases, and another one is applicable to general cases. We then present an algorithm for finding a primitive element and its minimal polynomial over $\mathbf{Q}$ by using these generating vectors.

5.1. Generating vectors

Let W be a vector subspace of $\mathbf{Q}^n$ which is determined by $\mathbf{Q}$-linear relations between $\alpha_1, ..., \alpha_n$, i.e.

$W = \{\bar{s} \in \mathbf{Q}^n | \alpha(\bar{s}) = 0\}$.

Since K is the splitting field of f, $K/\mathbf{Q}$ is a Galois extension. Then the set $\mathcal{G}$ of all distinct embeddings from K into $\mathbf{Q}$ is a Galois group of $K/\mathbf{Q}$. For each g in $\mathcal{G}$, α_i^g lies in $S = \{\alpha_1, ..., \alpha_n\}$. So g is considered as a permutation on the set of indices $\{1, ..., n\}$ and $\mathcal{G}$ is considered as a permutation group on $\{1, ..., n\}$. Then $\mathcal{G}$ acts on $\{1, ..., n\}$ faithfully.

We write α_{i^g} instead of α_i^g. Moreover, $\mathcal{G}$ can act on the set of coordinates of $\mathbf{Q}^n$ and so $\mathcal{G}$ can act on $\mathbf{Q}^n$ as follows.

For a vector $\bar{s} = (s_1, ..., s_n)$ and $g \in \mathcal{G}$, $\bar{s}^g = (s_{1^g}, s_{2^g}, ..., s_{n^g})$.

Then the following holds.

$\alpha(\bar{s}^g) = \alpha(\bar{s})^{g^{-1}}$.

Furthermore, by observing the action of $\mathcal{G}$ on W, we have the following lemma.

Lemma 5.1. *W is $\mathcal{G}$-invariant, i.e. $W^g = W$ for every $g \in \mathcal{G}$.*

Similarly as in Section 4, we define V and $V(g)$ for $g \in \mathcal{G} - \{1\}$. By Lemma 4.1, $\alpha(\bar{s})$ is not a primitive element of K if and only if $\bar{s}$ lies in V. Consider the case where $\bar{s}$ lies in $V(g)$ for some $g \in \mathcal{G}$, i.e. $\alpha(\bar{s})^g = \alpha(\bar{s})$. Then by the previous argument, it follows that

$\alpha(\bar{s} - \bar{s}^{g^{-1}}) = \alpha(\bar{s}) - \alpha(\bar{s})^g = 0$.

Hence $\bar{s} - \bar{s}^{g^{-1}} \in W$ and so $\bar{s}^g - \bar{s} \in W$. From this, V and $V(g)$ are expressed as follows.

$V = \{\bar{s} \in \mathbf{Z}^n | \bar{s}^g - \bar{s} \in W \text{ for some } g \in \mathcal{G} - \{1\}\}$,

$V(g) = \{\bar{s} \in \mathbf{Z}^n | \bar{s}^g - \bar{s} \in W\}$.

Now we are ready to give examples for generating vectors. First we consider special cases, and next we discuss the general case.

Case where $\mathcal{G}$ is doubly transitive

Consider the case where $\mathcal{G}$ acts on $\{1, ..., n\}$ doubly transitively, i.e. for each pair (i, j) of distinct elements in $\{1, ..., n\}$ there is some g in $\mathcal{G}$ such that $(i^g, j^g) = (1, 2)$. We note that we can test whether $\mathcal{G}$ is doubly transitive or not by examining whether $f(x)/(x - \alpha)$ is irreducible over $\mathbf{Q}(\alpha)$ or not, where α is an arbitrary root of f. Then by using results of representation theory of groups, we have the following theorem.

Theorem 5.2. *Let $\bar{s} = (1, 2, ..., n-1, 0)$. Then $\alpha(\bar{s})$ is a primitive element of K.*

Proof. Assume that $\alpha(\bar{s})$ is not a primitive element of K. Then there is an element g in $\mathcal{G} - \{1\}$ such that $\bar{s}^g - \bar{s} \in W$. Let $\bar{u} = \bar{s}^g - \bar{s}$. By the definition of $\bar{s}$, there are some coordinates of $\bar{u}$ which differ from each other, i.e. $u_i \neq u_j$ for some $i, j \in \{1, ..., n\}$. Let U be a subspace spanned by a set $\{\bar{u}^h | h \in \mathcal{G}\}$. Then U is a $\mathcal{G}$-invariant subspace of W. Moreover, let U^* be a vector space over $\mathbf{C}$ spanned by a set $\{\bar{u}^h | h \in \mathcal{G}\}$, i.e. $U^* = \mathbf{C} \otimes_{\mathbf{Q}} U$.

Now consider the actions of $\mathcal{G}$ on U and U^*. The action of G on $\mathbf{Q}^n$ is a permutation representation of $(\mathcal{G}, \{1, ..., n\})$ over $\mathbf{Q}$. Since $\mathcal{G}$ acts on $\{1, ..., n\}$ doubly transitively, by using the formula of the reciprocity of Frobenius, the representation of $\mathcal{G}$ on the extension $\mathbf{C} \otimes_{\mathbf{Q}} \mathbf{Q}^n \cong \mathbf{C}^n$ of $\mathbf{Q}^n$ is decomposed into two irreducible representations. One is the representation on the set $I = \{(a, a, ..., a) | a \in \mathbf{C}\}$, and the other is the representation on $J = \{(a_1, ..., a_n) | a_i \in \mathbf{C} \text{ and } \sum_{i=1}^{n} a_i = 0\}$. Since any representation of $\mathcal{G}$ on $\mathbf{C}^n$ is an irreducible representation or a sum of irreducible representations, U^* should be I, J or $I \bigoplus J$. But $\bar{u} = \bar{s}^g - \bar{s}$ does not lie in I. So $U^* = J$ or $I \bigoplus J$. From this, U^* contains J and so $\bar{a} = (1, -1, 0, ..., 0)$ lies in $U^* \cap \mathbf{Q}^n = U$. Hence $\bar{a}$ lies in W and $\alpha(\bar{s}) = 0$. This implies that $\alpha_1 = \alpha_2$ and a contradiction. Q.E.D.

We improve the above theorem.

Theorem 5.3. $\bar{s} = (1, 2, ..., n-1, 0)$ *is a generating vector of a primitive element.*

Proof. We prove Theorem 5.3 similarly as Theorem 5.2. Assume that $\bar{s}$ is not a generating vector. Then there is an element i in $\{1, ..., n\}$ such that $\alpha(\bar{s}(i))$ is not a primitive element of $\mathbf{Q}(\alpha_1, ..., \alpha_i)$. This implies that there is an element g in $\mathcal{G}$, which is not an identity embedding from $\mathbf{Q}(\alpha_1, ..., \alpha_i)$ into K, such that $\bar{s}(i)^g - \bar{s}(i) \in W$. By the proof of Theorem 5.2, if there are some coordinates of $\bar{s}(i)$ which differ from each other, then we get a contradiction. So $\bar{s}(i)^g - \bar{s}(i) \in I$, where I is defined in the proof of Theorem 5.2. By seeing $\bar{s}$, it follows directly that $\bar{s}(i)^g - \bar{s}(i) = 0$. Therefore, g fixes $1, ..., i$, i.e. g fixes $\alpha_1, ..., \alpha_i$. This implies that g is an identity embedding. Hence we get a contradiction. Q.E.D.

Case where n is prime

Similarly as in the previous case, $\bar{s} = (1, 2, ..., n-1, 0)$ is a generating vector for the case where n is a prime integer p. We use the following well-known lemma.

Lemma 5.4. (well-known) $\mathcal{G}$ *has a cyclic subgroup* C *of order* p. *Then* C *is transitive on* $\{1, 2, ..., p\}$.

By using this lemma, we have the following theorem.

Theorem 5.5. *Let* $\bar{s} = (1, 2, ..., p-1, 0)$. *Then* $\alpha(\bar{s})$ *is a primitive element of* K.

Proof. We prove Theorem 5.5 by similar arguments as in the proof of Theorem 5.2. Assume that $\alpha(\bar{s})$ is not a primitive element of K. Then there is an element g in $\mathcal{G}$ such that $\bar{s}^g - \bar{s} \in W$. Let $\bar{u} = \bar{s}^g - \bar{s}$. Then there are some coordinates of $\bar{u}$ which differ from each other. Let $\mathcal{C}$ be a cyclic subgroup of $\mathcal{G}$ of order p. Then $\mathcal{C}$ is transitive on $\{1, 2, ..., p\}$. Moreover, consider the subspace U spanned by $\{\bar{u}^h | h \in \mathcal{C}\}$. Then U is $\mathcal{C}$-invariant. By using similar arguments as in the proof of Theorem 5.2, to get a contradiction, we have only to show that $\dim U = p - 1$. By changing the order of indices, if necessary, there is an element c in $\mathcal{C}$ such that c is expressed by $(1, 2, ..., p)$. Now we consider the matrix M whose i-th row is a vector $\bar{u}^{c^{i-1}}$, i.e.

$$M = \begin{pmatrix} u_1 & u_2 & \dots & u_p \\ u_p & u_1 & \dots & u_{p-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_2 & u_3 & \dots & u_1 \end{pmatrix}.$$

Then the characteristic polynomial det(xI-M) of M is as follows.

det(xI-M)=$\prod_\omega(x - u_1 - \omega u_2 - ... - \omega^{p-1}u_p)$, where ω ranges all p-th roots of 1.

Then the multiplicity of 0 as an eigenvalue of M, is the number of p-th roots ω of 1 such that $u_1 + \omega u_2 + ... + \omega^{p-1}u_p = 0$. Since for any p-th root $\omega \neq 1$, there is no Q-linear relation between $1, \omega, ..., \omega^{p-1}$ except for $1 + \omega + ... + \omega^{p-1} = 0$, $u_1 + \omega u_2 + ... + \omega^{p-1}u_p$ cannot be 0 for $\omega \neq 1$. For $\omega = 1$, by the definition of $\bar{u}$ it follows that $u_1 + u_2 + ... + u_p = 0$. Hence the multiplicity of 0 as an eigenvalue of M is equal to 1 and this implies that the rank of M is equal to $p - 1$ and so the rank of U is equal to $p - 1$. Q.E.D.

By similar argument as in the proof of Theorem 5.3, we get the following theorem as an improvement of Theorem 5.5.

Theorem 5.6. $\bar{s} = (1, 2, ..., p - 1, 0)$ *is a generating vector.*

General case

Similarly as in Section 4.3, bounds b and c for roots of f are defined as follows.

Definition 5.2. Let b be a rational number which is greater than the absolute values $|\alpha_i|$'s, and let c be a positive rational number whose reciprocal is not greater than the absolute values of the differences of any two distinct roots of f. Moreover, let a be the smallest integer such that $a \geq 4bc$ and $a \geq 2$.

Then for a vector $\bar{a} = (1, a, a^2, ..., a^{n-2}, 0)$, we have the following as a corollary of Theorem 4.7 by the fact that K is generated by any $n-1$ distinct roots of f.

Corollary 5.7. $\bar{a} = (1, a, a^2, ..., a^{n-2}, 0)$ *is a generating vector.*

Hence we have the following algorithm.

Algorithm 5.1. (determining the generating vector)

Input: f an irreducible polynomial over $\mathbf{Q}$.

Output: $\bar{s}$ a generating vector.

n :=degree f;
if n is prime then return $(1, 2, ..., n-1, 0)$;
$g := f(x)/(x-t)$ over $\mathbf{Q}[t]/f(t)$;
h :=Res$_t(f(t), g(x-2t))$; (Norm of g)
factorize h over $\mathbf{Q}$;
if h is irreducible over $\mathbf{Q}$ then return $(1, 2, ..., n-1, 0)$;
compute b, c and a; (see Algorithm 4.2.)
return $(1, a, a^2, ..., a^{n-2}, 0)$

It is easy to incorporate the generating vector algorithm into the scheme of trial algorithm, for example, Trager's algorithm. We explain the new algorithm.

5.2. Algorithm using generating vectors for splitting fields

By incorporating generating vectors in Trager's algorithm, we obtain a non-trial algorithm, which we call Modified-Algorithm. In Trager's algorithms, for each i, we have to seek an integer s_i by trial such that $s_1\alpha_1 + \cdots + s_i\alpha_i$ is a primitive element of the intermediate field $K_i = \mathbf{Q}(\alpha_1, ..., \alpha_i)$, where $(s_1, ..., s_{i-1})$ is already determined vector giving a primitive element of the previous intermediate field $K_{i-1} = \mathbf{Q}(\alpha_1, ..., \alpha_{i-1})$. In Modified-Algorithm, integers s_i are determined without trial in advance. The remaining job is to compute the minimal polynomial of each primitive element.

One may think that the generating vector will give very large coefficients to the minimal polynomial, and that the advantage of avoiding trials will be lost. However,

there are cases where Trager's algorithm obtains, after a long series of trials and failures, the same generating vector as Modified-Algorithm gives. Consider the case where the Galois group $\mathcal{G}$ of $K/\mathbf{Q}$ is the symmetric group S_n of degree n. In this case, Algorithm 5.1 gives $(1, 2, ..., n-1, 0)$. On the other hand, Trager's algorithm obtains the same vector $(1, 2, ..., n-1, 0)$. This is concluded by the following lemma which can be proved easily.

Lemma 5.8. *For the case where $\mathcal{G} = S_n$, the following holds.*

For each $i \geq 2$, $s_1\alpha_1 + \cdots + s_i\alpha_i$ can not be a primitive element of $\mathbf{Q}(\alpha_1, ..., \alpha_i)$, if there exists a pair of integers (j, k), $1 \leq j \neq k \leq i$, such that $s_j = s_k$, where $\alpha_1, ..., \alpha_i$ are i distinct roots of f.

In Trager's algorithm, at each step for constructing K_i, candidates for s_i are chosen in ascending order from the set $\{1, 2, ...\}$. (Trager's algorithm really does so.) Since the already computed s_j's must be all distinct by Lemma 5.8, the primitive element of K_{i-1} must be $\alpha_1 + 2\alpha_2 + \cdots + (i-1)\alpha_{i-1}$ in Trager's algorithm. At this step, i trials are necessary to get $s_i = i$, since no candidate for s_i from the set $\{1, ..., i-1\}$ is admissible. Especially, at the final step for $i = n-1$, $n-1$ trials are unavoidable. This shows that the cost of this computation is $O(n)$ times larger than that of Modified-Algorithm. Thus, Modified-Algorithm improves the efficiency when the computed generating vector is relatively small.

Now, we show the complete algorithm.

Algorithm 5.2. (Modified-Algorithm)

Input: f an irreducible polynomial over $\mathbf{Q}$.

Output: g the minimal polynomial of a primitive element of the splitting field, and the representations of the roots of f by a primitive element.

Step 1. compute the generating vector $\bar{s}$;
Step 2. $i := 1$;
$g := f(x)$;
$R := \emptyset$;
$F := \{f(x;t)/(x-t)\}$ in $(\mathbf{Q}[t]/g(t))[x]$;
Step 3. $i := i + 1$;
For each $P(x;t)$ in F do
$Q_P(x)$:=$\mathrm{Res}_t(g(t), (s_i)^{deg\ P} P((x-t)/s_i;t))$;
factorize $Q_P(x)$ over $\mathbf{Q}$;
L_P :=set of all irreducible factors of Q_P;
$h_P(x)$:= element of max degree in L_P;
$L := \cup_P L_P$;
$h(x)$:=polynomial with max degree among $h_P(x)$;
Step 4. For each $T(x)$ in $L \setminus \{h(x)\}$ do
$V(x;t)$:=GCD$((s_i)^{deg\ P} P((x-t)/s_i;t), T(x))$ over $\mathbf{Q}[t]/g(t)$;
$U(x;t) := (1/s_i)^{deg\ V} V(s_i x + t;t)$;
If deg $U(x;t) = 1$ then
$W(t) := -$constant part of $U(x;t)$;
$R := R \cup \{W(t)\}$;
else $F := F \cup \{U(x;t)\}$;
$V(x;t)$:=GCD$((s_i)^{deg\ P} P((x-t)/s_i;t), h(x))$ over $\mathbf{Q}[t]/g(t)$;
$U(x,t) := (1/s_i)^{deg\ V} V(s_i x + t;t)$;
If deg $U(x;t) = 1$ then
$W(t) := -$constant part of $U(x;t)$;
$R := R \cup \{W(t)\}$;
Return g and R;
$u(x;t)$:=GCD$(g(x), V(t;x))$ over $\mathbf{Q}[t]/h(t)$;
$g(x) := h(x)$;
$v(t) := -$constant part of $u(x;t)$ over $\mathbf{Q}[t]/g(t)$;
$U(x;t) := U(x;v(t))$ over $\mathbf{Q}[t]/g(t)$;
$m(t) := (t - v(t))/s_i$;
$R := \{r(v(t)) | r(t) \in R\} \cup \{m(t)\}$;
$V(x;t) := U(x;t)/(x - m(t))$;
$F := \{U(x;v(t)) | U(x;t) \in F\}$;
If deg $V(x;t) = 1$ then
$W(t) := -$constant part of $V(x;t)$;
$R := R \cup \{W(t)\}$;
else $F := F \cup \{V(x;t)\}$;
Step 5. If $|R| = n$ then
Return g and R;
else goto Step 3

6. Concluding remarks

Our results are summarized as follows. An improved bound for the number of trials in the existing algorithms for the separable-type case is given. For the mixed-type case of the problem, t-base-trial method is proposed, and the number of trials is shown to be no more than that for the separable-type case. The notion of the generating vector is introduced, and based on which a non-trial method for extension fields is presented. The generating vector is also applied to splitting fields. The presented algorithms are faster than other existing methods in the stage where the primitive element and its minimal polynomial are computed. The total algorithm, however, usually includes the stage of representing adjoined elements by the obtained primitive element. To obtain the representation, our algorithms employ GCD computation over extension fields as others do. Since the cost of this stage is most dominating, the cost as a whole is not decreased so much. For splitting fields, a non-trial method is also presented. It is shown that our algorithm is faster than a typical existing algorithm with trials. The improvement is more remarkable than that for arbitrary extension fields, even if the representation stage is included. The remaining difficulty is that the cost of factoring polynomials over extension fields is also dominating as much as above GCD computation. This implies that our generating vector must be small as much as possible. In general, however, the generating vector tends to be large. These difficulties are left for further investigation.

Acknowledgements

The authors would like to thank the editor and referees for their valuable comments and suggestions.

This is part of the work in the major R&D of the Fifth Generation Computer Project, conducted under program set up by MITI.

References

Arnon, D. S., Sederberg, T. W. (1984). Implicit equations for parametric surfaces by Gröbner basis. Proceedings of the 1984 MACSYMA User's Conference, 431.

Collins, G.E. (1971). The calculation of multivariate polynomial resultants. J. ACM **19**. 515-532.

Feit, W. (1967). Characters of finite group. New York. W. A. Benjamin Publ..

von zur Gathen, J. (1984). Parallel algorithms for algebraic problems. SIAM J. Comput. **13**. 802-824.

Gianni, P., Trager, B., Zacharias, G. (1987). Gröbner bases and primary decomposition of polynomial ideals. preprint.

Landau, S. (1985). Factoring polynomials over algebraic number fields. SIAM J. Comput. **14**. 184-195.

Lang, S. (1976). Algebra. 8th ed. Reading, Massachusetts. Addison-Wesley.

Lenstra, A. K. (1983). Factoring polynomials over algebraic number fields Computer Algebra. Lecture Notes in Computer Science **162**. New York. Springer-Verlag. 245-254.

Loos, R. (1982a). Generalized polynomial remainder sequence. In: (Buchberger, B. et al., eds) Computer Algebra (Computing Supplement **4**). New York. Springer-Verlag. 173-187.

Loos, R. (1982b). Computing in algebraic extensions. In: (Buchberger, B. et al., eds) Computer Algebra (Computing Supplement **4**). New York. Springer-Verlag. 173-187.

Kobayashi, H. (1987). preprint (in Japanese).

Kobayashi, H., Moritsugu, S., Hogan, R. W. (1988). On solving systems of algebraic equations, preprint.

Mignotte, M. (1982). Some useful bounds. In: (Buchberger, B. et al., eds) Computer Algebra (Computing Supplement **4**). New York. Springer-Verlag. 259-263.

Trager, B. M. (1976). Algebraic factoring and rational integration. Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation.

van der Waerden, B. L. (1931). Moderne Algebra. Berlin. Springer-Verlag.

Weinberger, P. J., Rothschild, L. P. (1976). Factoring polynomials over algebraic number fields. ACM Trans. Math. Softw. **2**. 335-350.

Yokoyama, K., Noro, M., Takeshima, T. (1987). Computing primitive elements for extension fields. Research Report No.80. IIAS-SIS.